

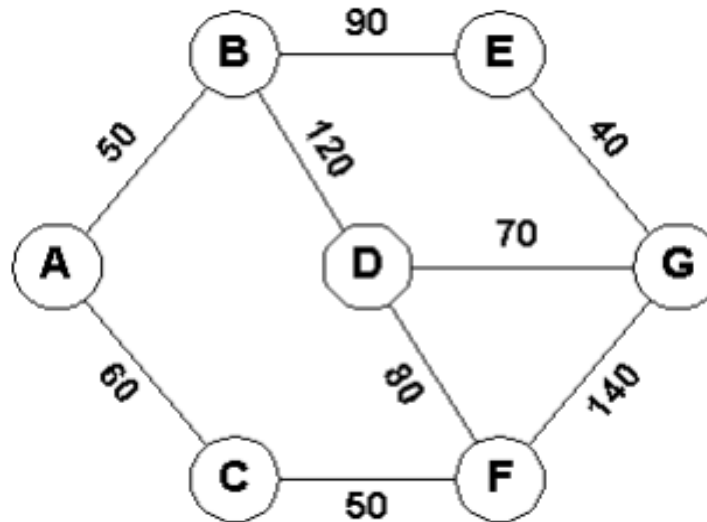
Audiophobia

Big-O Blue - Lecture 15: Prim

Tóm tắt đề bài

Tóm tắt đề bài

Cho một đồ thị các đường đi trong thành phố, mỗi con đường được cho biết mức cường độ âm thanh tính bằng decibel.



Yêu cầu: Ứng với mỗi truy vấn u, v , tìm ra con đường có mức âm thanh tối thiểu mà bạn có thể chịu đựng khi đi qua con đường đó.

Mô tả Input/Output

Input

Gồm nhiều test case, mỗi test case:

- Dòng đầu: C, S, Q tương ứng với số đỉnh, số cạnh và số truy vấn
- S dòng tiếp theo, mỗi dòng gồm 3 số c_1 c_2 d (đỉnh đầu, đỉnh cuối và mức âm của cạnh đó)
- Q dòng tiếp theo, mỗi dòng gồm 2 số c_1 c_2

Kết thúc tập tin input ứng với ba số $C = 0$, $S = 0$, $Q = 0$.

Output

Với mỗi truy vấn c_1 , c_2 : in ra **mức âm thanh tối thiểu**.

Nếu không có đường đi từ c_1 đến c_2 thì in ra **“no path”**

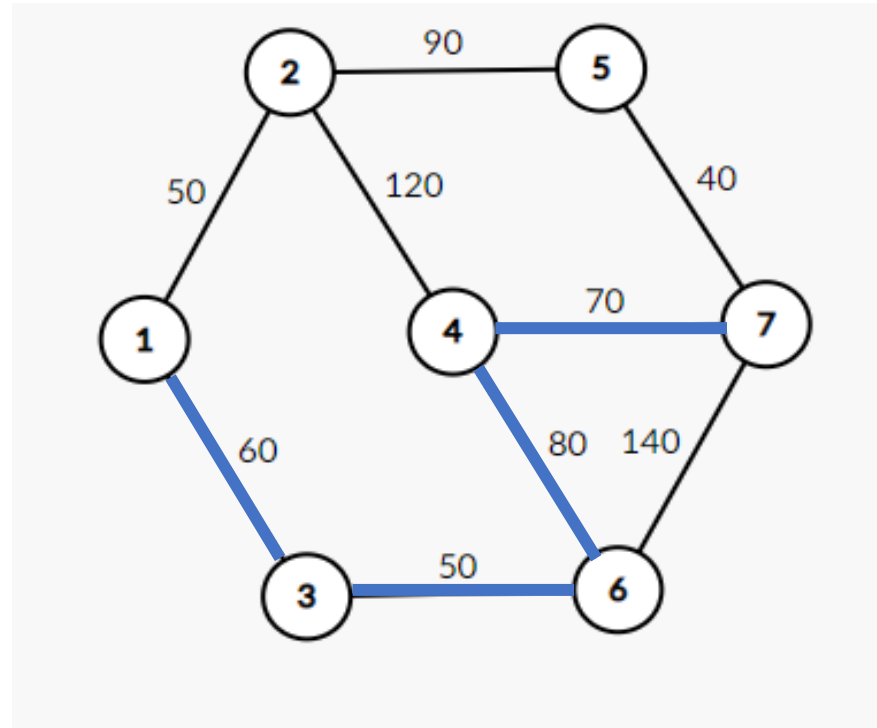
Giữa các test case có một dòng trống

Giải thích ví dụ

Ví dụ 1

Input

```
7 9 3
1 2 50
1 3 60
2 4 120
2 5 90
3 6 50
4 6 80
4 7 70
5 7 40
6 7 140
1 7
2 6
6 2
```



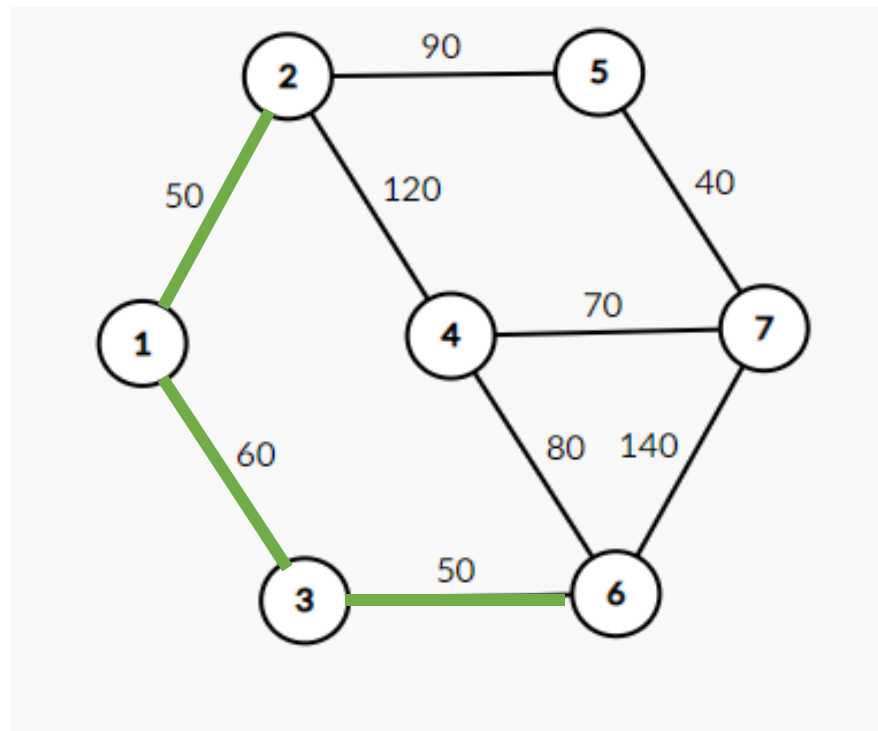
Output

Truy vấn (1,7): 80

Ví dụ 1

Input

```
7 9 3
1 2 50
1 3 60
2 4 120
2 5 90
3 6 50
4 6 80
4 7 70
5 7 40
6 7 140
1 7
2 6
6 2
```



Output

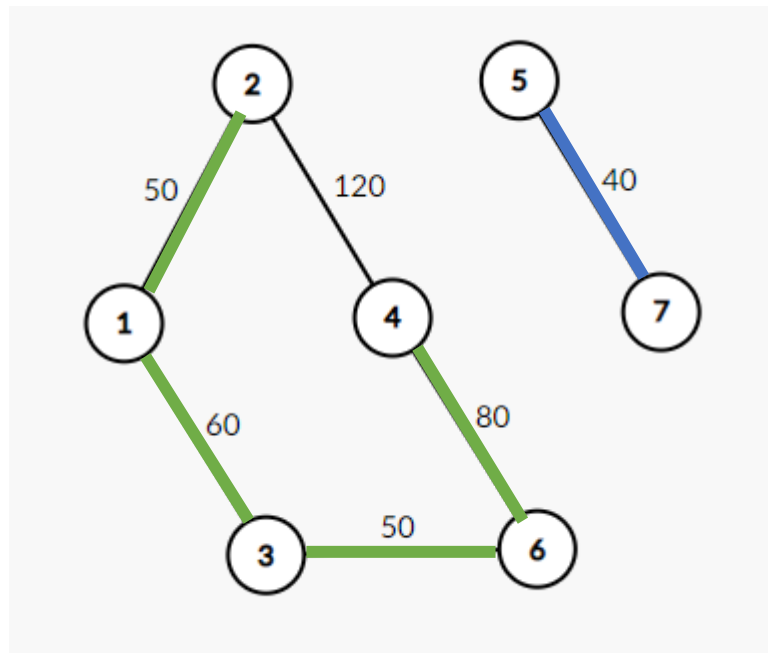
Truy vấn (2,6): 60

Truy vấn (6,2): 60

Ví dụ 2

Input

```
7 6 3
1 2 50
1 3 60
2 4 120
3 6 50
4 6 80
5 7 40
7 5
1 7
2 4
```



Output

Truy vấn (7,5): 40

Truy vấn (1,7): no path

Truy vấn (2,4): 80

Hướng dẫn giải

Nhận xét

Đường đi có mức âm tối thiểu sẽ nằm trên cây khung nhỏ nhất

⇒ Đối với một đồ thị bất kì, ta sẽ tìm cây khung nhỏ nhất trước (Prim).

⇒ Sau khi đã có cây khung nhỏ nhất, áp dụng BFS/DFS để tìm đường đi từ $c1 \rightarrow c2$.

⇒ Kết quả sẽ là cạnh có trọng số lớn nhất trên path từ $c1 \rightarrow c2$.

Lưu ý: đồ thị có thể là một rừng, tức có nhiều đồ thị đơn lẻ, nên ta phải tìm từng cây khung ứng với từng đồ thị.

Các bước giải

B1: Đọc C, S, Q và đọc vào đồ thị

B2: Tìm cây khung nhỏ nhất bằng thuật toán Prim. Nếu đồ thị là rừng \rightarrow tìm cây khung tương ứng với từng đồ thị. Lưu các cây khung vào một đồ thị mới mstGraph.

B3: Duyệt BFS/DFS trên mstGraph để tìm đường đi từ $c1 \rightarrow c2$. Trong path từ $c1 \rightarrow c2$, lưu cạnh có chi phí lớn nhất

B4: In kết quả

Độ phức tạp: $O(S \log C + Q * C)$,

- chi phí Prim $O(S \log C)$ với S là số cạnh, C là số đỉnh
- chi phí duyệt DFS với Q truy vấn: $O(Q * C)$ với Q là số truy vấn, C là số đỉnh ban đầu.

Mã giả

Mã giả

```
#main()
tc = 0
while true:
    read C, S, Q
    if C == 0 and S == 0 and Q == 0
        break

    graph = [[] for i = 0 .. C]
    dist = [INF for i = 0 .. C]
    visited = [False for i = 0 .. C]
    path = [-1 for i = 0 .. C]
    mstGraph = [[] for i = 0 .. C]

    for i = 1 .. S:
        read u, v, w
        push (v, w) to graph[u]
        push (u, w) to graph[v]
```

Mã giả

```
function prim(s, graph, dist, path, visited):  
    pq = PriorityQueue()  
    push (s, 0) to pq  
    dist[s] = 0  
    while pq not empty:  
        u = get top from pq  
        if visited[u] == true  
            continue  
        visited[u] = true  
        for (v, w) in graph[u]  
            if visited[v] == false and dist[v] > w:  
                dist[v] = w  
                path[v] = u  
                push (v, w) to pq
```

Thank you